Homework:

1. Calculate the number of integers in the decimal representation of n^2, n^3, 2^n, $n!$, n^n if $n = 100,1000,10,000,100,000,1,000,000$. Your answer should be in tabular form. You can make your life easier by using the decadic logarithm in Python. The number of digits in a number x is $\lceil \log_{10}(x)) \rceil$ unless the number is a power of 10, in which case you have to add one. $\lceil x \rceil$ is the ceiling of the number x and implemented in Python as math.ceil. Alternatively, you can convert the result into a string and then call len on the string. But be prepared to wait.

	n^2	n^3	2^n	$n!$	n^n
100	5	7	31	158	201
1000	7				
10000	9		3011		
100,000	11				500001
1,000,000	13			5565709	600001

2. Implement the abstract data type **Counter**. The ADT has operations increment, decrement, is_zero, and get_value. Its implementation has a single value per Counter object, originally 0. If you are decrementing a counter with zero value, than the result is still zero.This is a somewhat silly exercise, but you need to provide the following class methods:
   - `__init__`
   - `__repr__`
   - `__str__`
   - `get_value()`
   - `increment()`
   - `decrement()`
   - `is_zero()`

   Hand in the python code as a single python module with .py extension. Also implement the following test:

```
import random

my_counter = Counter()
for i in range(5):
   my_counter.increment()
print(my_counter.get_value(), 'should be 5')
for i in range(3):
   my_counter.decrement()
print(my_counter.get_value(), 'should be 2')
for i in range(4):
   my_counter.decrement()
for i in range(4):
   my_counter.increment()
print(my_counter.get_value(), 'should be 4')
random.seed(12345)
```

```
for i in range(100):
   if random.random()<0.5:
         Counter.increment()
   else:
         Counter.decrement()
print(Counter.get_value())
```